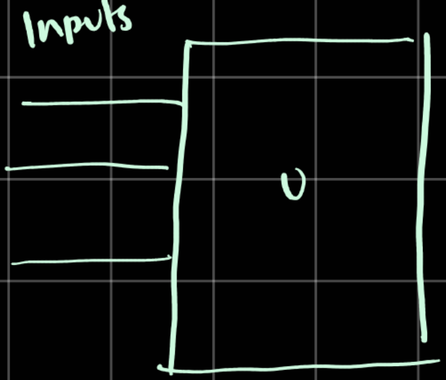


$\rightarrow \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \cdots \theta_n x_n$

$\downarrow$

o/p

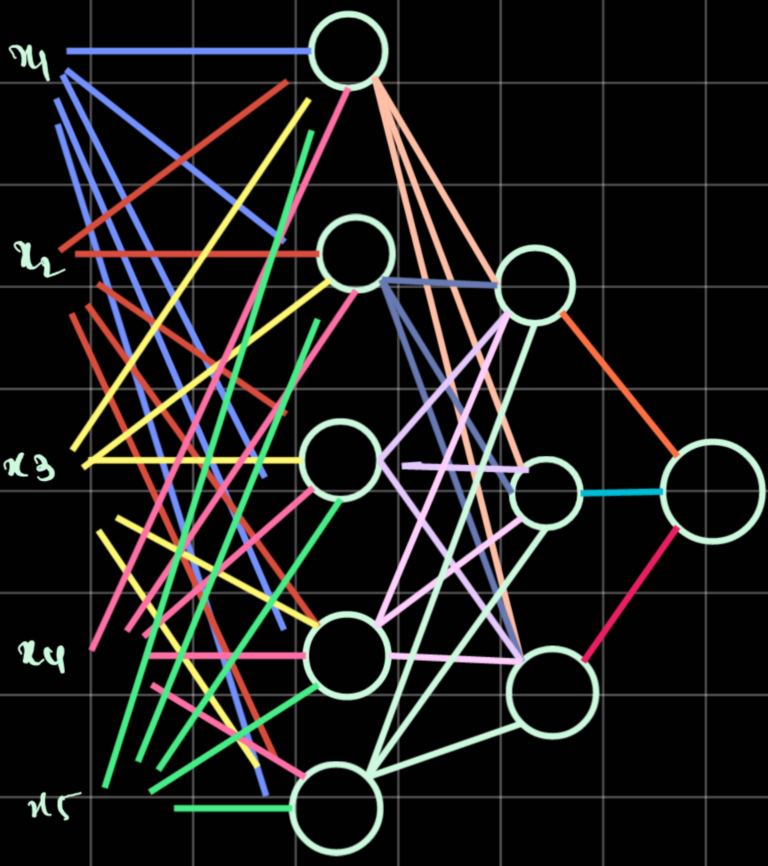
Activation function

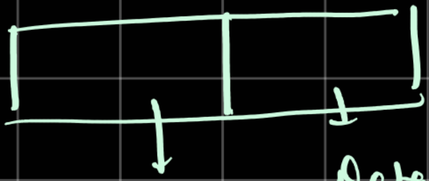
↳ Adds non linearity to a linear function

↳ $A(\theta_1 x_1 + \theta_2 x_2 + \cdots \theta_n x_n)$

Ex:- (sin (o/p) or cos (o/p))

Feed forward neural network

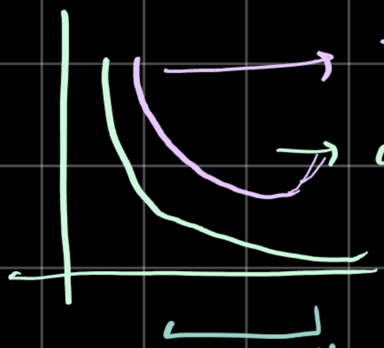




Data
↳ to test

Data
to train

epoch → if all data is trained once.



$J_{m,c}$

→ train loss increases after a threshold after a point

→ after multiple epochs the traing loss des.

↓
we stop epoching at a point.

elbow model.

∴

Data

1 2 | 3 4

train | test → to stop training

Data
train | test | validatation
↓
to check on unseen data

→ Normalization

→ Trying to stop overfitting & underfitting

Label → Yes/No (or) 0/1.

↳ But an actual function

↓

$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \cdots \theta_k x_k$?

can give 'R' as o/p

↓

to make this into 0/1

we use a sigmoid function. ←

$\sigma(\theta_0 + \theta_1 x + \theta_2 x_2 + \cdots \theta_k x_k) = 0$

↳ $\sigma = \dfrac{1}{|1 + e^{-u}|}$

∴ Loss function:?

$\frac{1}{n} \sum_{i=1}^{n} (\sigma^{f(x)} - y)^2$ → difficult to diff or Integrate

↓

we use a derivable Loss function ?

Negetive log likelihood loss function (NLL function).

⇒ $-y \ln(\underbrace{\sigma(f(x))}_{\text{probability of } y' \text{ being } 1 \text{ or } 0}) - (1-y) \ln(1 - \sigma(f(x)))$

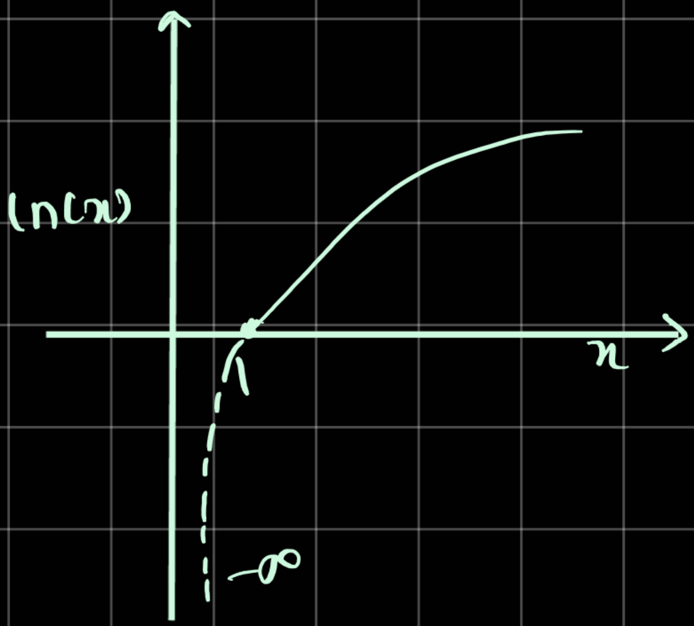
Let ideal function 'y' have pure o/p's 1 and 0.

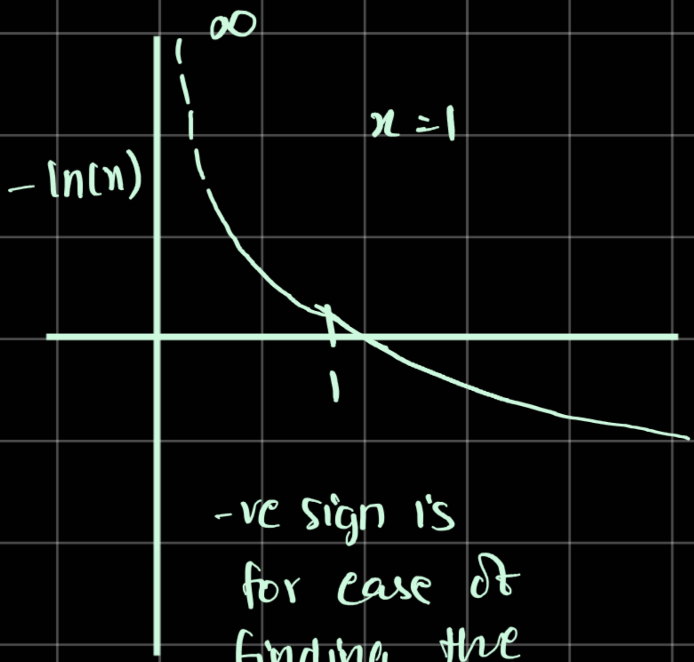
Here in Eq :- if probability : $\sigma(f(x))$

not probable : $1-\sigma(f(x))$

$\therefore$

$$\text{Loss} = \begin{cases} -\ln(\sigma f(x)) & y==1 \\ -\ln(1-\sigma f(x)) & y==0. \end{cases}$$





-ve sign is for case of finding the corresponding min.

$$\Rightarrow \frac{d\sigma(x)}{dx} = \sigma(x)(1-\sigma(x))$$